A REALISTIC NAVIGATION ALGORITHM FOR AN INTELLELIGET ROBOTIC WHEELCHAIR

H. Mahjoubi[1], F. Bahrami[2]

[1, 2] Control and Intelligent Processing Center of Excellence,
School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
e-mail: h.mahjoubi@ece.ut.ac.ir , fbahrami@ut.ac.ir

Abstract - **This paper introduces a novel navigation method for a robotic wheelchair. Based on observed characteristics of real user's motion, a realistic representation and navigation method is developed. The simple structure of the search space provided by the representation method accelerates the performance of the algorithm and thus, makes it suitable for real-time applications. A heuristical weighting approach is developed for the path planning module. This part is responsible for detection of an appropriate sub-goal within the sight of the wheelchair. Special bio-inspired potential fields are used to control the movement of the system toward this point. Upon facing a dynamic obstacle, an extra bio-inspired module will be applied to adjust the navigation task. The results of simulations in different environments have shown that the proposed method is highly robust and reliable.**

Keywords – **Assistive robotics, environment representation, dynamic obstacles, heuristical path planning, intelligent wheelchairs, obstacle avoidance.**

I. INTRODUCTION

In recent years, assistive robotics has considerably improved the quality of life for disabled people. Powered wheelchairs are the most common kind of such systems. While using theses devices, user does not need to push the wheels in order to move the wheelchair. In this case, one controls the movement using a peripheral device such as joystick. However, many people who use wheelchairs are unable to effectively control the movement of a typical powered wheelchair due to special/multiple disabilities or lack of experience.

During the past decade, there has been a considerable effort to develop a robotic wheelchair system that provides navigational assistance in different environments, which will allow its user to drive more efficiently. Wheelesley [1] is one of the early intelligent wheelchairs. This semi-autonomous robot can travel safely in an indoor environment using various sensors. Through a graphical interface, users can easily navigate the wheelchair by selecting a simple instruction. Each instruction represents a course of several navigational tasks which will be performed automatically. NavChair [2] is another good example of intelligent wheelchairs. This model has three modes of operation: (1) obstacle avoidance, (2) door passage, and (3) wall following. Based on the environmental surroundings, the control system automatically changes the mode to the most appropriate one. The TAO series [3] are another famous group of intelligent wheelchairs. In addition to the tasks performed by NavChair, these prototypes have two additional capabilities: (1) escape from a crowded environment, and (2) perform landmark based navigation. TinMan series [4] and Rolland [5] are two other examples with the ability of safe indoor navigation.

The majority of available intelligent wheelchairs are designed only for navigation in indoor environments and thus, they can only avoid static obstacles. MAid (Mobility Aid for Elderly and Disabled People) [6] is one of the rare examples that can safely navigate in crowded environments with dynamic obstacles (e.g., a railway station). An extension of Wheelesley is another successful example of outdoor intelligent wheelchairs [7]. Here, it is assumed that obstacles in the middle of a sidewalk are likely to be moving. In addition, objects moving toward the wheelchair will move around it, and objects moving away from the chair will move faster. With these assumptions, the behavior for dealing with obstacles is to slow if an obstacle appears in 1.5 to 3 meters from the wheelchair and to stop while facing a closer one (0.6 to 1.5 meters from the wheelchair). Although this method is shown to be partly successful, it is highly unrealistic and in a considerable number of experiments, the users prefer to interfere with the task of navigation around a dynamic obstacle.

In addition to the problem of avoiding dynamic obstacles, there is another problem with current intelligent wheelchairs. The main goal of such systems is to reduce the number of necessary commands for any maneuver while performing it as safe as possible [1]. These commands must be relatively simple so that anyone can control the wheelchair using them. Many automatic wheelchairs have not completely fulfilled this characteristic. In this work, we have used a two level approach in order to achieve this goal. The *higher* level includes the detection of an appropriate sub-goal (direction of movement) and adjustment of the speed. These are the only two parameters that the user needs to control. If there is no input, the system itself may adjust these parameters. In the *lower* level, the process of navigation, obstacle avoidance and other delicate maneuvers will be controlled automatically. To make the performance of this part as realistic as possible, we have tried to extract the main features of the movement of a wheelchair controlled by a typical user. These features are then used in the design of the lower level. A realistic movement helps the user to feel safer and thus, he/she does not need to interfere with the task of navigation in many occasions. In fact, the necessary commands will be limited to those concerning the determination of a certain goal and those that imply a specific adjustment in the direction/speed of the movement.

The details of this approach will be discussed in section II. In section III, the performance of the presented method in several simulated environments is examined and the corresponding results are presented. Section IV includes the discussions and section V concludes the paper.

II. METHODOLOGY

A. Environment Representation

The representation method to be used is previously presented by the authors in [8]. There, it has been discussed that in a conventional representation method such as *composite space map* [9], there is a direct relationship between the number of the members of the search space and the precision of the representation. However, a large search space will result in slower performance of the path planner and thus, it can not be used for real-time applications. The proposed representation method [8] has the benefit of reduced search space while representing the environment as accurate as possible. Thus, it can be used in a real-time application such as navigation of an intelligent wheelchair.

Based on the fact that a wheelchair usually travels over flat surfaces, the environment can be assumed to be two dimensional. Thus, the sensory input of the surroundings of the wheelchair can be represented in the form of a 2D binary image in which the black parts represent the obstacles (as in Fig. 1.a).

While moving, humans usually consider the obstacles within a certain range from their current location. Based on this fact, we have chosen a range of 3 meters for the input data. Within this field, the edges of the obstacles influence the shape of the trajectory [8]. In addition, humans usually consider a safety margin while moving around an obstacle [9]. Our representation method is based on these two characteristics. In the beginning, those edges of the obstacles which are in direct sight of the wheelchair (a reference point: the center of the input image) will be detected (Fig. 1.b). Then, the morphological operand "*dilation*" [10] will be applied on the detected edges. This leads to the expansion of the edges with a selected safety margin (0.6 meters). Fig. 1.c shows the result of this step. The grey areas are the expansions of the obstacles. In the last step, the boundaries of these expansions will be found and their corner points will be detected (Fig. 1.d). Knowing which corner points are in direct sight of each specific corner point, one can simply use these points to rebuild the original shape of the surrounding environment.

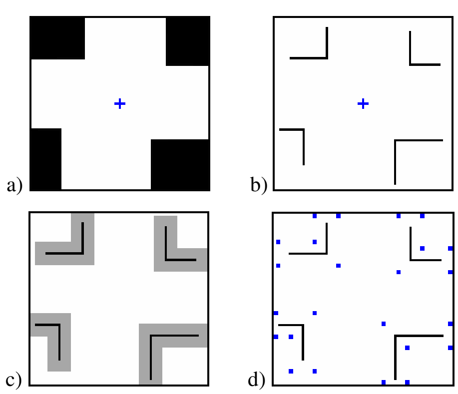

Fig. 1. (a) The local surroundings of a certain point. (b) The edges of the obstacles within the local frame which are in the direct sight of the reference point. This is what the system really senses from its surroundings. (c) Expansion of the edges in Fig. 1.b. (d) Detection of the corner points (representation of the local frame).

B. Path Planning Algorithm

To begin the path planning procedure, the start and destination configurations must be known. Here, the coordinates of the goal are specified with respect to that of the reference point on the simulated wheelchair. Once this is done, the path planner will find an appropriate sub-goal.

To introduce the effect of the goal, we will find the point where the connecting line of the reference and the goal positions crosses over the boundary of the current local frame (3 meters from the reference point). This point (D) will be added to the group of previously found corner points. All points – including the reference – are treated as nodes. Starting from the reference point, a cost will be calculated for every node n which is in direct sight of the current node (n-1):

$$f(n) = [g(n) + h(n)][1 + \frac{c \times |\omega_o(n)|}{300}] \quad (1)$$

where $f(n)$ is the calculated cost for node n, $g(n)$ the Euclidian distance between nodes n and n-1, $h(n)$ the Euclidian distance between nodes n and D, and $\omega_o(n)$ the deviation between the heading direction of the modelled wheelchair and the direction of node n with respect to the reference node (in degrees). Parameter c is assumed to be 1 if node n is in direct sight of the reference node, otherwise it is zero. It has been shown [11] that humans usually concentrate on the center of their field of view (FOV). Thus, the weight of direction is designed so that the chance of selection for the points closer to the center of FOV increases.

After calculating the cost of all possible nodes, the one with the lowest cost will be indexed as the current node and the above mentioned steps will be repeated for it. Once this node lies on the boundary of the local frame, this phase will be finished. Then, the last point in the generated series which is in direct sight of the original reference will be selected as the sub-goal. In case of encountering a local minima (e.g., a dead end), the controller switches the wheelchair to a modified wall following mode. The details are discussed in [12].

The model has also the ability to store the sensory input image of visited locations in order to memorize the global features of the environment. Once a global minima occurs (e.g., visiting a point for the second time), this memory is used along with the local frame to provide the model with a strategy to avoid previously selected paths (global path planning).

C. Motion Control and Navigation

In [13] a pair of bio-inspired potential fields has been introduced that successfully simulated the trajectories which were produced by healthy human subjects. Using the same approach for wheelchair users, we derived the following formulas for these forces:

$$attraction = 7.5 \times \Delta\psi_a (e^{-0.4 \times d_a} + 0.4) \quad (2)$$

$$repulsion = 5000 \times \Delta\psi_r e^{(-6.5 \times \frac{\pi}{180}|\Delta\psi_r| - 0.0016 \times d_r)} \quad (3)$$

where $\Delta\psi_a$ is the deviation between the heading of the model and the direction of the attractor (the sub-goal) with respect to

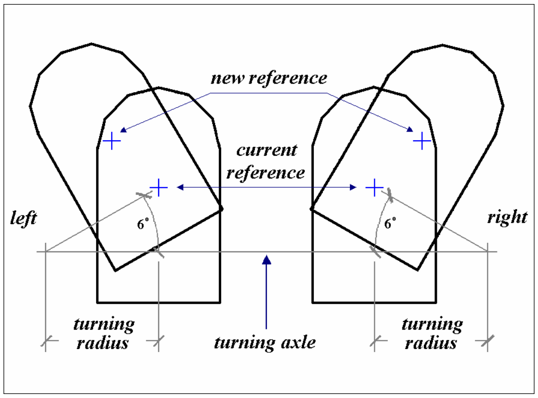

Fig. 2. The turning procedure of the simulated robotic wheelchair.

the reference point (in degrees), and d_a is the Euclidian distance between the attractor and the reference position (in meters). $\Delta\psi_r$ is the deviation between the heading of the model and the direction of the repeller (the closest obstacle point to the reference point) with respect to the reference point (in degrees), and d_r is the Euclidian distance between the repeller and the reference position (in meters). Using these two forces, a new heading will be determined. Depending on the desired speed (normally 1.5 m/s) and time step for each frame (0.1 s), the reference point – the wheelchair – will be moved along this direction (15 cm).

Since the model has turning constraints, the above steps will result in improper paths when the deviation of attractor from the heading of the model is more than 15 degrees. In this case, the model will be allowed to select an appropriate turning center (turning radius is assumed to be 55 cm). Then, it will turn around this point with a step of 6° (angular speed is assumed to be 60 °/s) in order to face the sub-goal. The reference point will be changed correspondently (Fig. 2).

After moving to a new reference point, a new input image will be provided and the phases of part A to C will be repeated. This continues until the final destination is reached.

D. Avoiding Mobile Obstacles

Mobile obstacles can be detected by comparing the last two sensory input images. Thus, we can estimate their speed and direction of movement and find out whether there is a chance of collision with one of them. A collision condition can be coded using seven parameters:

- Is the path around left blocked by other obstacles (Y or N)?
- Is the path around right blocked by other obstacles (Y or N)?
- Relative speed of the colliding obstacle
- Approach direction of the colliding obstacle
- Colliding obstacle's distance to collision point
- Wheelchair's distance to collision point
- Desired travel direction

In [14], this coding scheme is used to encode a set of collected data from the walking of healthy subjects (including the situation and the reaction of individuals). These reactions include: reduction/increase of speed, go around right, go around left, and no-action. In simulations, a naive Bayes classifier compares every encountered situation with training data. Then, the corresponding reaction to the most similar set of training data will be used to handle that situation.

Here, we have used a two-layer neural network to perform this classification. The inputs are the seven parameters used to encode an encountered situation. The outputs are the five possible reactions. All neurons have linear activation function. In training phase, for every input data only one of the outputs, which corresponds to the selected reaction, was assumed to be 1 and the rest were set to 0. While simulating a movement, the corresponding response to the highest output is used to adjust the overall reaction. A sample result is shown in the next section (Fig. 5).

III. Results

All parts of the described algorithm are implemented in MATLAB 7.0 on a 3 GHz, P4 PC with 512 MB onboard RAM. The wheelchair is assumed to be an 80 cm × 110 cm nonholonomic four-wheeled structure. The safety margin of the obstacles is assumed to be 60 cm. The normal speed and the time step are set to 1.5 m/s and 0.1 s, respectively.

To determine the performance speed of the algorithm in each step, we have used the profile function of MATLAB. The largest time required for simulation of one step was 52 ms which was observed near obstacles with curved boundaries. It is also worthy to note that on average, 38 ms of this time was used for representation of the local frame.

Fig. 3 to 5 show several simulated trajectories in different situations.

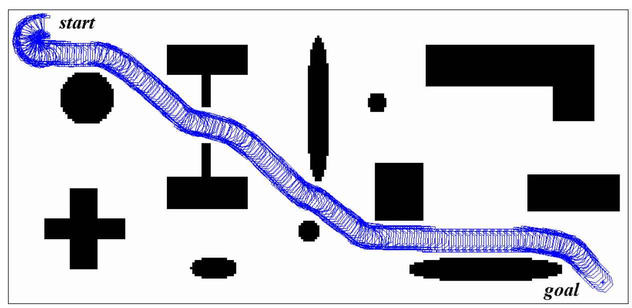

Fig. 3. The generated path in a static environment with both circular and rectangular obstacles.

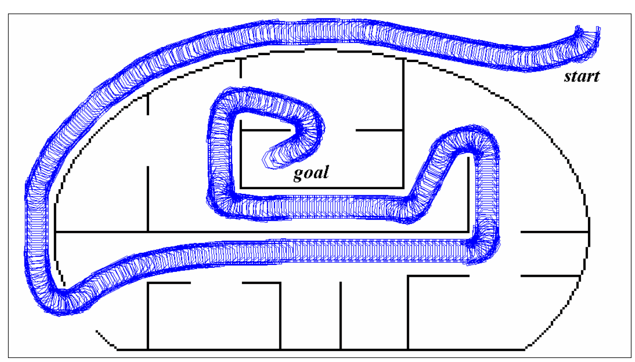

Fig. 4. By adapting the wall following algorithm with our representation method [12], the simulated system can effectively navigate through maze-like environments.

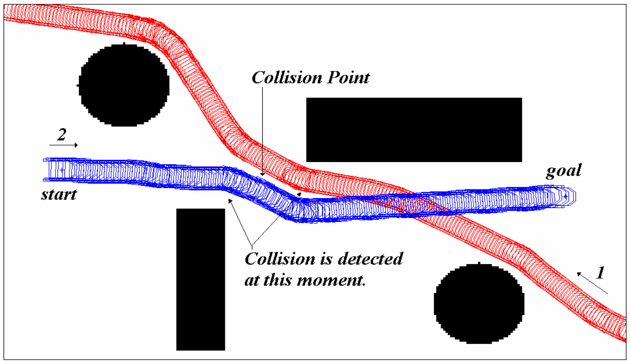

Fig. 5. The first trajectory (1) represents the path of a mobile obstacle. The second trajectory (2) is the path of the simulated wheelchair which has avoided the obstacle by going around right.

IV. Discussion

By solely defining a goal, the navigation algorithm of section II can automatically navigate the simulated wheelchair through different environments. The simulation results suggest that the generated paths are safe and realistic. In addition, the fast performance of the algorithm ensures that it can properly operate in daily environments.

V. Conclusion

Representation of the environment with the corner points of expanded obstacles' boundaries is the basic idea of section II.A. This method produces a considerably small search space which in turn results in fast performance of the path planner. The heuristical path planner of section II.B. and the motion controller of section II.C are responsible for navigation of the model in each represented frame. Wherever possible, we have tried to use bio-inspired models to make the final results more realistic.

The presented path planner is shown to be robust for navigating unknown static environments. In addition, by updating the input data in every step, it can detect and handle sudden changes in the static configuration of the model's surroundings. However, mobile obstacles must be treated separately. To do this, we have introduced another bio-inspired model in section II.D. After encoding the situation, a trained classifier analyzes the circumstances and chooses the best possible response. This choice is then used to adjust the overall reaction.

Some typical simulation results have been presented in section III along with a brief statistic on the execution time of each step. These results prove that the presented navigation algorithm is both robust and fast. The paths are smooth and safe; however, the user should be able to control their shapes with simple high level commands. In a wheelchair which uses this work as a platform, the user may suggest a sub-goal by pushing a joystick towards it. After controlling its accessibility, the navigation module guides the system to that sub-goal or informs the user about his/her inappropriate choice. User may also adjust the speed with the amount of pressure on the joystick. If there are no inputs from the user, the system will automatically determine these parameters.

References

[1] H.A. Yanco, A. Hazel, A. Peacock, S. Smith, H. Wintermute, "Initial report on Wheelesley: a robotic wheelchair system," In Proceedings of the Workshop on Developing AI Applications for the Disabled, Montreal, Canada, 1995.

[2] S.P. Levine, D.A. Bell, L.A. Jaros, R.C. Simpson, Y. Koren, and J. Borenstein, "The NavChair assistive wheelchair navigation system," *IEEE Transactions on Rehabilitation Engineering*, Vol. 7, No. 4, pp. 443-451, 1999.

[3] D.P. Miller, "Assistive robotics: an overview," In Mittal et al. eds., Assistive technology and AI, LNAI-1458, Berlin, Springer-Verlag, pp. 126-136, 1998.

[4] P.D. Nisbet, "Who's intelligent? Wheelchair, driver or both? In Proceedings of IEEE International Conference on Control Applications, Anchorage, AK, pp. 760-765, 2002.

[5] A. Lankenau, T. Röfer, and B. Krieg-Bruckner, "Self-localization in large-scale environments for the Bremen autonomous wheelchair," In Freksa and et al. eds., Spatial Cognition III, LNAI-2685, Berlin, Springer-Verlag, pp. 34-61, 2003.

[6] E. Prassler, J. Scholz, and P. Fiorini, "A Robotic Wheelchair for Crowded Public Environments," *IEEE Robotics and Automation Magazine*, Vol.8, No. 1, pp. 38-45, 2001.

[7] H.A. Yanco, "Development and testing of a robotic wheelchair system for outdoor navigation," In Proceedings of the Conference of Rehabilitation Engineering and Assistive Technology Society of North America, RESNA Press, 2001.

[8] H. Mahjoubi, F. Bahrami and C. Lucas, "Path planning in an environment with static and dynamic obstacles using genetic algorithm: a simplified search space approach", In Proceedings of IEEE World Congress on Computational Intelligence (WCCI), Vancouver, Canada, 2006.

[9] S. Bandi, and D. Talmann, "Space discretization for efficient human navigation," Computer Graphic Forums, Vol. 17, No. 3, pp. 195-206, 1998.

[10] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, 2^{nd} Edition, Prentice Hall, New Jersey, 2002.

[11] M. A. Hollands, A. E. Patla, and J. N. Vickers, "Look where you're going: gaze behavior associated with maintaining and changing direction of locomotion," *Exp. Brain Res.*, Vol. 143, pp. 221-230, 2002.

[12] H. Mahjoubi, and F. Bahrami, "A novel heuristic approach to real-time path planning for nonholonomic robots," (in press: accepted in the 3^{rd} International Conference on Autonomous Robots and Agents (ICARA), Palmerston North, New Zealand, 12-14 December, 2006).

[13] B.R. Fajen and W.H. Warren, "A dynamical Model of visually guided steering, obstacle avoidance, and route selection", *International Journal of Computer Vision*, Vol. 54 (1/2), pp. 13-34, 2003.

[14] R. A. Metoyer , and J.K. Hodgins, "Reactive pedestrian path following from examples," In Proceeding of the 2nd International Conference on Computer Animation and Social Agents, New Brunswick, New Jersey, USA, pp. 149-156, 2003.